Soketi i transportni protokoli

Transportni sloj je četvrti sloj u OSI modelu, zadužen za prikupljanje podataka korisnika iz aplikacijskog sloja i njihovu pripremu za dalji transport kroz mrežu, tačnije, prvenstveno za transport kroz mrežni sloj. Odgovornost transportnog sloja se ogleda u dužnosti da korisničke podatke *transportuje* od izvora do odredišta.

Veza između transportnog i mrežnog sloja se ogleda u tome da transportni sloj pruža komunikaciju između procesa na datom uređaju (host), dok mrežni sloj pruža komunikaciju između uređaja (hosts). Takođe, deo mrežnog sloja je i IP (Internet Protocol), koji nam zapravo omogućava ovu komunikaciju između uređaja. Zato se može reći da su transportni protokoli odgovorni isključivo za komunikaciju od procesa do procesa.

Napomena

Često se srećemo sa pojmom IP adresa. IP adresa je numerička vrednost dodeljena svakom uređaju povezanom na računarsku mrežu koji koristi IP (Internet Protocol) kao način komunikacije. Trenutno su u opticaju dve verzije IP-ja: stariji, IPv4 (adrese ove verzije se pišu u formatu: 172.16.254.1) i noviji: IPv5 (IP adrese ove verzije se pišu u formatu 2001:db8:0:1234:0:567:8:1). Ako je IP adresa vidljiva samo u našoj računarskoj mreži, onda je reč o lokalnoj adresi, a ako je ta adresa vidljiva (i može joj se pristupiti) van naše mreže, sa interneta, onda je reč o javnoj IP adresi.

Neke od funkcija transportnog sloja

- Transportni sloj nam pruža okruženje (interfejs) za pristup mrežnih aplikacija mreži.
- Transportni sloj nam pruža okruženje (interfejs) za prihvatanje podataka iz različitih aplikacija na trenutnoj mašini i slanje tih podataka ka odredišnoj mašini. Isti ovaj proces se odvija i na odredišnoj mašini.
- Transportni sloj podržava mehanizme za dostavu podataka preko mreže bez gubitaka.
- Transportni sloj pruža mehanizme za otkrivanje grešaka, kontrolu toka prenosa i ponovno slanje nedospelih podataka (paketa).

Portovi i soketi

Soketi (priključci, engl. sockets) nam omogućavaju komunikaciju između dva različita procesa na istoj mašini ili različitim mašinama. Iz ugla programera, soketi izgledaju i ponašaju se vrlo slično kao fajl deskriptori nižeg nivoa, jer podržavaju koncepte kao što su upis i čitanje. Soket se sastoji iz dva dela razdvojena karakterom dve tačke (:) – IP adrese i broja porta.

U primeru 127.0.0.1:80:

- 127.0.0.1 je adresa internet protokola (IP);
- 80 je broj porta (proizvoljno u rasponu od 0 do 65535).

Broj ulaza (port number) je 16-bitni broj koji se nalazi u zaglavlju (headeru) protokola unutar transportnog sloja. Takođe, portovi moraju biti jedinstveni brojevi u okviru protokola. Na primer, mogu postojati dva ista porta u UDP i TCP protokolu, ali ne mogu postojati dva ista broja u TCP protokolu.

Broj porta može zauzeti maksimalno 16 bita, što nam daje 65535 brojeva na raspolaganje.

Portovi su podeljeni na opsege:

- brojevi od 0 do 1023 – već poznati portovi, koje je alocirala organizacija IANA (Internet Assigned Numbers Authority), dodeljeni serverima na korišćenje;
- brojevi od 1024 do 49151 – registrovani portovi; ovi brojevi nisu alocirani od strane IANA-a ali se mogu registrovati; dakle, sa IP adresom tražimo željeni uređaj na mreži, a broj porta nam omogućava da komuniciramo sa željenim procesom na prethodno pronađenom uređaju.
- Brojevi od 49152 do 65535 – ovi brojevi su dinamični portovi i nisu dodeljeni, kontrolisani niti registrovani. Koriste se za privremene ili privatne portove.

Lista portova se može naći na zvaničnom linku IANA-e.

Vrste soketa

Postoje četiri vrste soketa koje su nam dostupne generalno (a sve četiri vrste su nam dostupne i u Pythonu). Od te četiri vrste soketa, prve dve se češće koriste, a druge dve ređe. Preporučuje se da procesi komuniciraju između soketa istih vrsta, ali ovo nije pravilo.

- Datagram soketi (engl. datagram sockets) – Isporuka poslatih podataka nije garantovana. Ovi soketi su *connectionless* – bez uspostavljanja direktne veze. Dakle, nije nam potrebna već ostvarena konekcija da bismo slali podatak – paket se generiše sa informacijama o destinaciji i kao takav šalje. Ovi soketi koriste UDP protokol, o kome će biti reči kasnije u lekciji.
- Soketi toka (engl. stream sockets) – Isporuka poslatih podataka je garantovana. Ako podatke pošaljemo u 1,2,3 redosledu, u istom redosledu će i pristići. Ovi soketi za komunikaciju koriste TCP protokol. U slučaju da isporuka iz nekog razloga nije moguća, pošiljalac šalje grešku primaocu.
- Neprerađeni soketi (engl. raw sockets) – Pružaju korisnički pristup svim protokolima iz nižih slojeva. Nisu namenjeni korisnicima generalno, već služe za razvijanje novih protokola. Datagramski su orijentisani.
- Sekvencirani soketi (engl. sequenced packet sockets) – Slični su soketima toka, pa im je tako prvenstveno potrebno uspostavljanje konekcije. Ovi soketi dalje omogućavaju rad sa Sequence Packet Protocolom (SPP) ili Internet Datagram Protocolom (IDP).

Transportni protokoli

Među glavne protokole transportnog sloja spadaju TCP (Transmission Control Protocol) i UDP (User Datagram Protocol). Unutar njih su implementirani drugi protokoli (a koji su deo viših slojeva OSI modela) kao što su HTTP, HTTPS, VoIP i slični. Pošto je IP kao takav nepouzdan (jedino što IP garantuje je ostvarivanje konekcije; ne garantuje siguran i uređen pristup paketa), uloga UDP-ja i TCP-ja je da prošire funkcionalnost IP-ja. Takođe, i UDP i TCP pružaju proveru integriteta prispelih podataka. TCP i UDP koriste portove kako bi odredili kojem procesu na trenutnoj mašini poslati potrebni podatak.

Zanimljivost

Datagram (kovanica nastala od reči *data* i *telegram*) jeste jedinica mere, kao što je i paket, ali se razlikuje od paketa u tome što datagram ne zahteva potvrdu da je primljen.

Pitanje

Soket je sastavljen od:

- IP adrese
- porta
- **IP adrese i porta**

Objašnjenje:

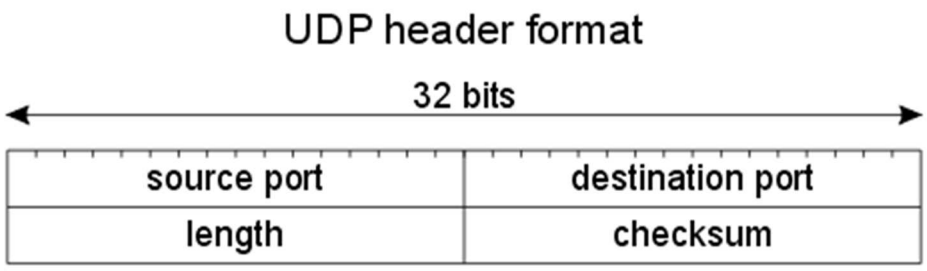
Tačan odgovor je da je soket zapravo kombinacija IP adrese i porta.

UDP Protokol

UDP (User Datagram Protocol) je protokol koji ne zahteva prethodno ostvarivanje konekcije. Nema ugrađenu kontrolu grešaka (ne prijavljuje greške, ali može proveriti da li one postoje) i neće ponovo poslati zagubljene pakete. Zbog ovih nedostataka, ovaj protokol je dosta brži nego TCP, ali se zato smatra nepouzdanim. Pogodan je u situacijama kada nam je potrebno da konstantno i brzo primamo podatke, kao što je na primer video poziv ili gledanje video-klipa uživo preko interneta. Kao i svi ostali protokoli, i UDP sadrži zaglavlje (header) koje čine određena, unapred definisana polja, kao i deo za podatke koje se prenose (payload). Zaglavlje UDP protokola čine:

- source port (16 bits) – broj porta odakle se podatak (paket) šalje;
- destination port (16 bits) – broj porta gde paket treba ispostaviti;
- length (16bits) – dužina čitavog UDP datagrama uključujući i zaglavlje i podatke koji su deo trenutnog paketa;
- checksum (16bits) – kontrolni zbir čitavog datagrama (uključujući i zaglavlje i podatke koji su deo trenutnog paketa); opcioni je i koristi se za verifikaciju integriteta primnljenog podatka i zaglavlja; takođe, ovaj deo zaglavlja u sebi sadrži još neke podatke, tačnije 12 bajta takozvanog *pseudo-headera*, nasleđenih iz IP datagrama, koji se ponajviše koriste radi kalkulacije kontrolnog zbira. Tih 12 bajta iz IP datagrama čine:

- IP source address (4 bajta) – IP adresa uređaja odakle se paket šalje;
- IP destination address (4 bajta) – IP adresa uređaja kome se paket šalje;
- UDP Length (2 bajta) – dužina UDP datagrama koja je nasleđena iz IP zaglavlja;
- protocol (1 bajt) – polje iz IP zaglavlja rezervisano za protokol (za UDP je to 17);
- zero (1 bajt) – polje koje sadrži samo nulu.



Slika 2.1. Tabelarni prikaz UDP zaglavlja[1]

TCP Protokol

Kako bi TCP (Transmission Control Protocol) funkcionisao, potrebno je prvo uspostaviti konekciju. Podržava traženje i prijavljivanje grešaka.

Neki od aplikativnih protokola koji se baziraju na TCP-u su HTTP i HTTPS, koji su rezervisani za sajtove, SMTP i POP, koji su rezervisani za slanje mejlova, kao i FTP – za slanje fajlova.

Tri osnovna principa TCP protokola su:

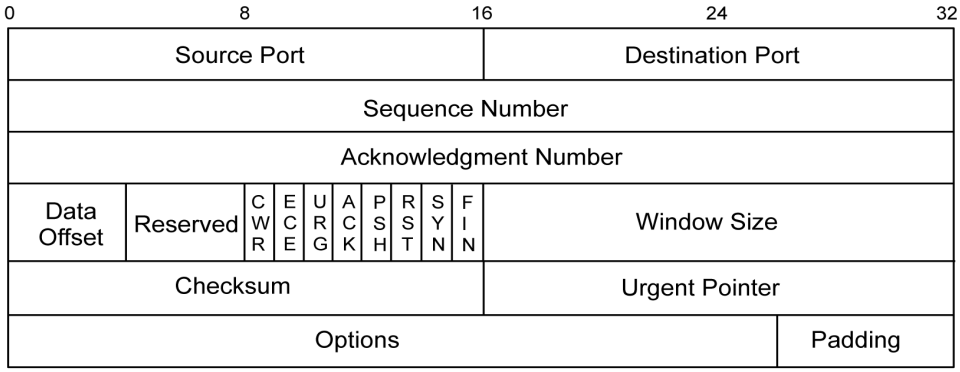
- obezbeđivanje pouzdane dostave segmenata po zadatom redosledu;
- obezbeđivanje kontrole toka kako bi primalac mogao da primi sve segmente;
- provera ispravnosti primljenih segmenata (upravljanje greškama) i zahtevanje ponovnog slanja oštećenih segmenata.

TCP razlaže podatke koje prima iz aplikativnog sloja u manje delove – segmente. Ti segmenti se numerišu pre slanje IP-ju, koji ih dalje enkapsulira u pakete. TCP prati broj poslatih paketa; ako ne dobije potvrdu o dospeću tih paketa u određenom roku, pretpostaviće da su paketi izgubljeni i poslati ih opet. Takođe, TCP proverava svaki segment zbog mogućih nastalih promena (upravljanje greškama), što je omogućeno poljem *checksum* u zaglavlju. Kao i UDP, a i ostali protokoli, i TCP koristi sokete kako bi naznačio kom procesu i mašini treba dostaviti paket. Kako TCP može primiti veliki broj segmenata koji se moraju proveriti svaki pojedinačno, potrebno je kontrolisati količinu segmenata koji se mogu primiti. Ovo definiše pošiljalac, kroz polje *window size*, u kom se određuje koliko segmenata primalac može primiti dok ne pošalje potvrdu prijema.

[1] https://medium.com/@orhunp/tr-project-review-hexene-localvpn-de1a82d0c0dd

Spisak polja u zaglavlju je sledeći:

- source & destination port (2 x 16 bits) – TCP port pošiljaoca i primaoca;
- sequence number (32 bits) – broj prvog bajta podatka u payload delu segmenta;
- acknowledgment number (32 bits) – sledeći bajt koji se očekuje, što znači da je primalac primio sve bajtove do njega;
- flow control window (16 bits) – veličina kontrolnog prozora, definiše koliko segmenata primalac može primiti dok ne pošalje potvrdu prijema;
- data offset (4 bits) – koristi se za lociranje početka i kraja payload dela segmenta; treba imati na umu da TCP ne mora da sadrži nikakve podatke osim zaglavlja;
- urgent pointer (16 bits) – ako postoje urgentni podaci, ovo polje nagoveštava njihovu bajtsku lokaciju; kako terminali rade slično kao soketi (barem kada je reč o čitanju i pisanju), može se uzeti primer slučaja kada korisnik pošalje komandu Ctrl+C terminalu; tom kombinacijom tastera želimo da prekinemo i završimo trenutni proces u terminalu – u tom slučaju, Ctrl+C bi bio urgentni podatak;
- flags (6 bits) – ovo polje se sastoji od šest jednobitnih indikatora:
    - urgent pointer (URG) – ako je vrednost ovog indikatora 0, polje urgent pointer se ignoriše, u suprotnom ukazuje na njegovo postojanje;
    - acknowledgment valid (ACK) – indikator koji ukazuje na to da li je polje *acknowledgment number* validno; jedini slučaj kada ono nije validno je tokom trostrukog usaglašavanja (handshake) koje se dešava na početku konekcije;
    - rest (RST) – indikator koji se koristi za brzi prekid konekcije; svrha je signaliziranje grešaka;
    - push (PSH) – ako je ovaj indikator pristupan, to je nagoveštaj da primalac treba momentalno da preda trenutne segmente sloju iznad;
    - synchronization (SYN) – koristi se za iniciranje nove konekcije;
    - finish (FIN) – koristi se za prekid konekcije.



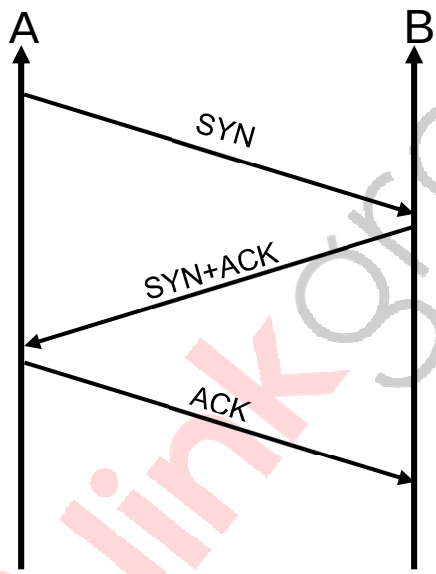
Slika 2.2. Tabelarni prikaz TCP zaglavlja[2]

[2] http://intronetworks.cs.luc.edu/current/html/tcp.html

Trostruko usaglašavanje (3-way handshake)

Ovaj način ostvarivanja konekcije je svojstven TCP protokolu. Recimo da je A klijent, a B server – onda ovo usaglašavanje izgleda ovako:

- Klijent A šalje serveru B segment sa postavljenim indikatorom SYN (SYN=1). ACK indikator je i dalje 0 (ACK=0).
- Server B odgovara sa sopstvenim paketom, koji takođe ima postavljen SYN indikator. U odgovoru, ACK indikator je postavljen na 1.
- Klijent A odgovara serverskom SYN indikatoru sa sopstvenim ACK indikatorom (ACK klijenta je sada postavljen na 1).



Slika 2.3. Dijagram trostrukog usaglašavanja

Ovakav način komunikacije se ostvaruje samo na početku i bez njega ne može početi prenos podataka.

Razlike između TCP-a i UDP-a

Razlike između ova dva protokola možemo prikazati u tabeli:

Kategorija	TCP	UDP
Naziv	Transmission Control Protocol	User Datagram Protocol
Protokol	Potrebna uspostava veze	Nije potrebna uspostava veze
Bezbednost	Prijavljuje i traži greške pomoću checksuma	Samo proverava da li postoje greške uz pomoć checksuma, ali ih ne prijavljuje
Slanje podataka	Sporo	Brzo
Primena	Email, FTP	VoIP

Tabela 2.1. Tabelarni prikaz razlika između UDP-a i TCP-a

Detaljnija analiza performansi ovih protokola može se videti na ovom linku.

Rezime

- Osnovni transportni protokoli transportnog sloja su UDP i TCP.
- Soketi nam omogućavaju komunikaciju između dva različita procesa na istoj mašini ili različitim mašinama.
- Soket čine dva dela: IP adresa i port (primer: 127.0.0.1:80).
- Najviše korišćeni tipovi soketa su stream sockets i datagram sockets.
- Port brojeve možemo proizvoljno birati, ali je preporuka da se nalaze u opsegu od 49152 do 65535.
- UDP (User Datagram Protocol) je protokol koji ne zahteva prethodno ostvarivanje konekcije i nema ugrađenu prijavu grešaka, ali je zato brži od TCP-a.
- TCP (Transmission Control Protocol) je protokol koji zahteva prethodno ostvarivanje konekcije pomoću trostrukog usaglašavanja i podržava prijavljivanje i traženje grešaka u segmentima, ali je zato sporiji od UDP-a.